



Office européen des brevets



(11)

**EP 1 043 699 A1**

(12)

**EUROPEAN PATENT APPLICATION**

(51) Int Cl.<sup>7</sup>: **G07D 5/00**, **G07D 7/00**

(21) Application number: 00302916.2

(22) Date of filing: 06.04.2000

(72) Inventor: **Billington, Gregory John  
Reading, Berkshire RG6 3BS (GB)**

(30) Priority: 07.04.1999 GB 9907951

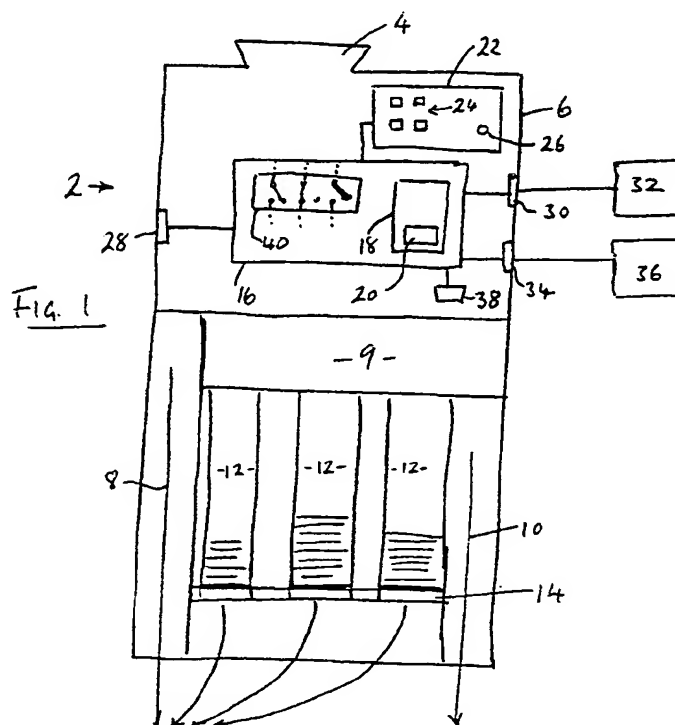
(74) Representative: **Burke, Steven David et al**  
**R.G.C. Jenkins & Co.**  
**26 Caxton Street**  
**London SW1H 0RJ (GB)**

(71) Applicant: **Mars Incorporated**  
**McLean, Virginia 22101-3883 (US)**

**(54) Money validation**

(57) A money validator, for example a coin validator, stores a boot program and a validation program. The boot program is run on power-up of the validator, checks for the presence of a validation program and transfers control to the validation program if one is found. Otherwise, the boot program continues to run, and includes

a communications routine to allow installation of a validation program. The validation program can transfer control to the boot program to permit data to be transferred using any one of a plurality of different communications ports, and this data can be used to re-configure the validator, for example by changing stored acceptance criteria.



## Description

**[0001]** This invention relates to methods and apparatus for validating units of money, and to methods for configuring and re-configuring such apparatus. The invention will be described primarily in the context of coin validators, but is also applicable to other apparatus such as banknote validators.

**[0002]** Electronic coin validators are often microprocessor controlled. The microprocessor either contains or is connected to a memory circuit which contains program code and data which are used in the validation of coins.

**[0003]** It is known to use replaceable memory circuits to permit the program of the validator to be upgraded. It is also known to use memory circuits which store alterable contents, so that the operation of the validator can be adjusted. For example, it is known to alter data representing acceptance criteria used to determine whether an inserted coin is a valid coin of a particular denomination. This allows the validator to be re-configured, or reprogrammed, to handle different denominations of coins from those which the validator was originally configured to accept. Alterable data may be stored in a separate memory circuit to facilitate alterations, such as an EAROM (electrically alterable read only memory).

**[0004]** To change the memory contents, it is necessary for the validator to have means permitting suitable connections to be made to the memory to allow the contents to be altered by external apparatus. This external apparatus may directly change the contents of the memory by means of this connection. In other known arrangements a communications port is provided for the external apparatus to communicate with the microprocessor for various purposes, including the reading-out of various operational parameters of the program. This port can additionally be used to instruct the microprocessor to alter the contents of the memory circuit.

**[0005]** Coin validators also have communication ports for other purposes. For example, there may be a port for communicating with a host vending machine in which the coin validator is located, a port for communication with other transaction equipment, such as a banknote validator, or a port permitting transaction data to be downloaded from the coin validator.

**[0006]** Various aspects of the invention are set out in the accompanying claims.

**[0007]** According to one aspect of the invention, a control means, which may include a microprocessor, for a money validator is operable to receive data from any selected one of a plurality of communication ports, and to use the received data to replace memory contents used in the validation operation performed by the validator. Thus, re-configuring of the validator is facilitated, because the re-configuration operation can be performed using the most suitable port, having regard for example to the speed of communication, physical access to the port, etc. The re-programming may take

place using a different port from that (if any) used in the original factory configuration of the validator.

**[0008]** According to a further aspect of the invention, a control means for a money validator has memory means (which may be a single circuit or multiple circuits) having first contents comprising a boot program and second contents defining a validation operation and comprising a validator program, the boot program being operable independently of the presence of the validation program and including a communications routine, the boot program being arranged to be executed on initiation of the operation of the validator, being selectively operable to transfer control to the validation program, and also being capable of using the communications routine to receive externally-supplied data and to use this data to replace at least some of the second memory contents. Preferably, the validation program is responsive to at least one command to relinquish control to the boot program; in this way, the boot program can be arranged automatically to relinquish control on initialisation, e.g. power up, so that validation operations can commence, but nevertheless provision is made so that the boot program can regain control in order to permit a communication operation to take place in order to re-configure the validator.

**[0009]** Preferably, these aspects of the invention are combined in a single embodiment.

**[0010]** The re-configuration can be carried out in order to alter acceptance criteria. Alternatively or additionally, the re-configuration operation can be carried out to alter at least some of the program code forming the validation program.

**[0011]** Preferably, the boot program is operable to scan a plurality of communication ports of the validator, to recognise that one of the ports is being used, e.g. by detecting data on that port, and thereafter to select that port and establish a communication operation using the selected port.

**[0012]** Preferably, the boot program is arranged to check for the presence of a validation program before relinquishing control after the operation of the validator has been initiated. Preferably, the validation program is operable to relinquish control to the boot program on receipt of a specific external command. Preferably, different commands can be provided using, e.g. a message on a communication port and/or an external switching means.

**[0013]** Preferably the boot program and the application program occupy respective areas of a memory map, and preferably the communication program is arranged such that the area occupied by the boot program cannot be overwritten by received data.

**[0014]** The invention also extends to a method of re-programming a validator in accordance with the invention.

**[0015]** An arrangement embodying the invention will now be described by way of example with reference to the accompanying drawings, in which:

Fig. 1 schematically shows a coin validator in accordance with the invention; and

Fig. 2 is flowchart of the operation of the validator.

**[0016]** Referring to Fig. 1, a coin validator 2 comprises a hopper 4 for receiving coins which are directed to an acceptor unit 6. The acceptor unit is arranged to perform testing of inserted coins. Any that are deemed invalid are directed to a reject path 8, which returns the coins to the customer. Acceptable coins are directed by a sorter 9 either to a cashbox (not shown) via a cashbox path 10, or to storage regions 12, each of which comprises a tube for receiving a respective denomination of coins. Coins from the tubes can be selectively dispensed using a dispenser unit 14.

**[0017]** The acceptor 6 has a control board 16 which is coupled to the various operational parts of the validator, including the coin sensors (not shown), the dispenser 14, various gates (not shown) in the sorter 9 which control the routing of the coins and further sensors, for example sensors to indicate when the tubes 12 are full. The circuit board carries a microprocessor 18 which is connected to a flash (EARAM) memory 20. The memory 20 may be formed in the same integrated circuit unit as the microprocessor 18, or may be one or more separate circuit elements.

**[0018]** The circuit board 16 is also connected to an input output unit 22 which has a keypad 24 and an LED indicator 26.

**[0019]** The control board 16 is also connected to four serial ports. One serial port 28 is intended to allow servicemen to interrogate the circuit board to obtain a history of transactions carried out by the validator. If desired, the port 28 may be in the form of an infra-red receiver transmitter for enabling communication with a similarly-equipped port on a downloading device carried by the service engineer.

**[0020]** The coin validator is housed in a vending machine (not shown), and a second of the ports, 30, is connected to a control board 32 of the vending machine.

**[0021]** A third port 34 is connected to a bill validator 36 also housed within the vending machine.

**[0022]** A fourth port 38 is provided for communication with a detachable programming unit.

**[0023]** Preferably, the port 28 is accessible without opening the vending machine, but the ports 30, 34 and 38 are only accessible by opening the vending machine. The ports 30, 34 and 38 are preferably of standard types. For example, the port 30 may comply with (i) the Multi-Drop Bus (MDB), which transmits data using the protocol in the NAMA "Internal Multi-Drop Bus Interface Standard", (ii) the Executive Vending Machine Interface (Protocol A) of Mars, Inc., or (iii) the BDV standard BDV001. The port 34 preferably comprises the MDB peripheral standard. The port 28 may comply with the DEX or DDCMP communications standard according to the European Vending Association Data Transfer Standard, or be a standard printer port. The port 38 is preferably

an HII port, connected to the circuit board 16 via a bus corresponding to that disclosed in EP-A-584097. All these standard types of ports are well known to those skilled in the art of coin validators.

**[0024]** In prior art arrangements, the port 38 is used for re-configuring validators. For example, it is possible using the port 38 to communicate with the microprocessor 18 in order to instruct the microprocessor to replace stored data representing acceptance criteria with further data transmitted via the port 38. Any data transmitted from the port 38 complies with a known protocol, which involves transmitting a command and data.

**[0025]** The circuit board 16 carries, or is connected to, electrical contacts which can be selectively bridged by a detachable connector 40. By selectively attaching the connector 40, it is possible to provide commands to the validator, the nature of the command depending upon which contacts are bridged by the connector. In the preferred embodiment, the connector is a plug having contacts mating with those on the validator, the contacts on the plug being selectively interconnected using a wire loom. The connector is carried by a service engineer and may be used to instruct the validator to accept a re-programming operation. The connector can be fitted only by opening the vending machine and partly disassembling the coin validator (in particular by unfastening and pivoting the acceptor unit 6 away from the remainder of the validator).

**[0026]** Referring to Fig. 2, this illustrates the operation which the validator carries out in response to programs stored in the flash memory 20. The memory map of the flash memory 20 contains a lower-address area and an upper-address area. The lower-address area contains a program routine which is executed on power-up of the microprocessor 18. This area contains a boot program which performs all the operations shown in Fig. 2 except for those shown within the program block 220. The upper-address area contains a validation program and associated data, which are used in the validation operation performed by the coin validator. The routines performed by the program code in the upper-address area are contained within the block 220.

**[0027]** Following power-up, at step 240, the boot program proceeds to an initialisation routine at step 242. Various initialisation operations are carried out here, including causing the LED 26 to display a red indication. Then, at step 244, the program examines the contents of the upper-address memory to determine whether an application program is present in that area. If so, the program proceeds to step 246. This part of the program checks the contents of the upper-address memory area to determine whether the application program should be deemed valid. This can be done in a number of ways, including performing processing of the stored contents to compare the results of the processing with stored checksums.

**[0028]** If no valid application is found in the upper-address memory area, the boot program proceeds from

step 244 or 246 to step 248. Thus, step 248 is reached either in response to an external instruction from a service engineer authorised to open the vending machine, or if no valid application program is present on power-up. The step 248 involves scanning all four of the communication ports 28, 30, 34 and 38 to determine whether any data is present on those ports. This routine is repeatedly performed until data is found, when the program proceeds to step 250. At this step, the LED 26 is caused to provide an alternating red/green display. The port on which the data has been found is selected as the port used for a communication operation.

**[0029]** Otherwise, assuming that the application is deemed valid, the boot program relinquishes control to the validation program 220. The validation program at step 220 includes a main validation routine 210 which will not be described in detail as it may be very similar to those executed by current validators, and which will be known to those skilled in the art. However, the validation program also checks for certain conditions, at step 212, 214 and 216, and if any of those conditions are met, the validation program relinquishes control back to the boot program, so that the program execution proceeds from the appropriate one of the steps 212 to 216 to step 248.

**[0030]** At step 212 the validation program checks to see whether a connector 40 of an appropriate configuration has been fitted.

**[0031]** At step 214, the validation program checks to determine whether a "reprogram" command has been received from a communication port. Preferably the program checks the reprogramming port 38, but it may additionally check one or more of the other ports.

**[0032]** At step 216, the validation program determines whether the keypad 24 has been operated in a predetermined way, which constitutes a reprogramming command.

**[0033]** Although the flowchart of Fig. 2 suggests that these three checking operations are performed in sequence, as part of a main program loop involving the validation operation, this is not essential. For example, the checking operations can be performed in response to interrupts.

**[0034]** In operation, a service engineer will open the vending machine and then connect a portable re-configuration device (which may be in the form of a standard PC-type computer with a standard communications port) to a selected port of the validator, if necessary unplugging the connection to port 30 or 34. The port may be selected according to convenience, taking into account the type of connectors available to the engineer, the speed of the port (as some ports may be faster than others), ease of access, etc. The engineer then instructs a reprogramming operation by either fitting the connector 40, operating the keypad in a predetermined way or issuing a command via a communications port. This results in control of the operation being relinquished by the validation program 220 to step 248 of the boot pro-

gram.

**[0035]** Then, at step 252, the program repeatedly executes a routine to check for a received command from a selected port. When that command has been received, the program proceeds to step 254 and then either to step 256 or to step 258 depending upon whether a valid application is present in the upper memory area. Assuming that a valid application is present, the program proceeds to step 256; this can only have been reached if the valid application has relinquished control to the boot program. At step 256, the LED 26 is caused to display a flashing amber colour. Otherwise, at step 258, the LED 256 is caused to alternate between green and amber colours. Step 258 will be reached only if no valid application was found on power up. Thus, the LED indicates whether or not the communications operation which is about to start will upgrade an existing application.

**[0036]** The program then proceeds to step 260 to determine the command which has been sent on the selected communication port.

**[0037]** There then follow a number of steps 262, 264, 266 and 268, which are carried out in order to check whether the received command corresponds to one of four types of commands. In practice, there would probably be more than four possible commands, in which case as indicated by path 269 additional steps to those shown in Fig. 2 would be provided to check for these further commands.

**[0038]** Step 262 checks for a "run validation program" command. If this has been provided, then the boot program relinquishes control to the application program at step 220.

**[0039]** Otherwise, at step 264, the boot program checks to determine whether the received command is a command to download data. If so, at step 270, a communication operation is performed on the selected communications port. This may use any standard communications routine, such as the XMODEM protocol.

**[0040]** The data received during this communication operation is stored in the flash memory, in regions of the upper memory address area determined by the parameters sent over the communications port with the download command. These contents can therefore overwrite part or all of the validation program stored in the upper memory area, or part or all of the data used by the validation program and also stored in the upper memory address area. Typically, this data may form acceptance criteria for the coins to be validated by the validator.

**[0041]** Following the download operation, the program proceeds to step 260 again, in order to obtain the next command. If this is a "run application" command, the program proceeds from step 262 to the application program at step 220, so that the re-configured validation program can be run.

**[0042]** The other commands checked for at steps 266, 268, etc. result in operations performed at steps 272, 274, etc. One command may for example be a request

for the boot program to send over the selected port a message indicating the maximum communications speed for that port. Another command may be an instruction to the boot program to change the set communication speed for the selected port to a particular value. These commands allow the speed of communication to be altered as desired by the equipment coupled to the selected communications port.

**[0043]** Thus, an engineer can cause a communication operation to take place to download data from his portable reprogramming equipment. The engineer then issues a "run application" command to re-start the re-configured validation operation, and then disconnects his equipment.

**[0044]** It will be appreciated from the above that the validation program 220 can be completely replaced using this technique. The boot program is so arranged that it does not permit overwriting of the lower memory-address area. Nevertheless, it is possible to upgrade the boot program in the following manner. First, the validation program is replaced by a new program corresponding to the boot program except without any limitation on the memory address areas to which data can be written. Then the original boot program relinquishes control to the newly-downloaded program. This then performs a communication operation to download a new boot program which is stored in the lower memory address area. Then control is relinquished to this newly-downloaded boot program. At that point, the original (or a new) validation program can be downloaded by the upgraded boot program.

**[0045]** It is well known to take measurements of coins and apply acceptability tests to determine whether the coin is valid and the denomination of the coin. The acceptability tests are normally based on stored acceptability data. One common technique (see, e.g. GB-A-1 452 740) involves storing "windows", i.e. upper and lower limits for each test. If each of the measurements of a coin falls within a respective set of upper and lower limits, then the coin is deemed to be acceptable. The acceptability data could instead represent a predetermined value such as a median, the measurements then being tested to determine whether they lie within predetermined ranges of that value. Alternatively, the acceptance data could be used to modify each measurement and the test would then involve comparing the modified result with a fixed value or window. Alternatively, the acceptance data could be a look-up table which is addressed by the measurements, and the output of which indicates whether the measurements are suitable for a particular denomination (see, e.g. EP-A-0 480 736, and US-A-4 951 799). Instead of having separate acceptance data for each test, the measurements may be combined and the result compared with stored acceptance data (cf. GB-A-2238152 and GB-A-2 254949). Alternatively, some of these techniques could be combined, e.g. by using the acceptance data as coefficients (derived, e.g. using a neural network technique) for combining the

measurements, and possibly for performing a test on the result. A still further possibility would be for the acceptance data to be used to define the conditions under which a test is performed (e.g. as in US-A-4 625 852).

**[0046]** It is known to use statistical techniques for deriving the data, e.g. by feeding many items into the validator and deriving the data from the test measurements in a calibration operation. It is also known for the validator to have an automatic re-calibration function, sometimes known as "self-tuning", whereby the acceptance data is regularly updated on the basis of measurements performed during testing (see for example EP-A-0 155 126, GB-A-2 059 129, and US-A-4 951 799).

**[0047]** Normally, the acceptance data produced by the calibration operation is characteristic of the specific type of item to be validated. However, it is alternatively possible for the data to be independent of the properties of the item itself, and instead to be characteristic of just the validation apparatus (e.g. to represent how much the apparatus deviates in its measurements from a standard) so that this data in combination with further data representing the standard properties of an item are sufficient for validation.

**[0048]** The acceptance criteria stored in the upper memory area of the above-described embodiment may take any of the forms indicated above, and it will be appreciated that the present invention provides a convenient way of altering such criteria.

## Claims

1. A money validator having a control means which is operable to receive data from any selected one of a plurality of communication ports of the validator, and to use the received data to replace memory contents used in the validation operation performed by the validator.
2. A money validator having a control means including memory means having first contents comprising a boot program and second contents defining a validation operation and comprising a validation program, the boot program being operable independently of the presence of the validation program and including a communications routine, the boot program being arranged to be executed on initiation of the operation of the validator, being selectively operable to transfer control to the validation program, and also being capable of using the communications routine to receive externally-supplied data and to use this data to replace at least some of the second contents.
3. A money validator as claimed in claim 2, wherein the boot program is arranged to check for the presence of a validation program before transferring control after the operation of the validator has been

initiated.

validation operation.

4. A money validator as claimed in claim 2 or claim 3, wherein the validation program is responsive to at least one command to transfer control to the boot program. 5
5. A money validator as claimed in claim 4, wherein the validation program is operable to transfer control to the boot program on receipt of a predetermined message from a communication port of the validator. 10
6. A money validator as claimed in claim 4 or claim 5, wherein the validation program is operable to transfer control to the boot program in response to an external switching means. 15
7. A money validator as claimed in any one of claims 2 to 6, wherein the first and second contents occupy respective areas of a memory map. 20
8. A money validator as claimed in claim 7, wherein the communications routine is arranged such that the area occupied by the boot program cannot be overwritten by received data. 25
9. A money validator as claimed in claim 8, wherein the communications routine is operable to use the externally-supplied data to write a second communications routine into the second contents area of the memory means and to transfer control to the second communications routine, the second communications routine being operable to receive further externally-supplied data and to use this to overwrite at least part of the boot program so as to upgrade the boot program. 30
10. A money validator as claimed in any one of claims 2 to 9, wherein the control means is operable to receive the externally-supplied data from any selected one of a plurality of communication ports of the validator. 35
11. A money validator as claimed in claim 1 or 10, wherein the control means is operable to scan a plurality of communication ports of the validator, to recognise that one of the ports is being used, and in response thereto to select that port and establish a communication operation using the selected port. 40
12. A money validator as claimed in any preceding claim, wherein the control means is operable to use received data to replace stored acceptance criteria. 45
13. A money validator as claimed in any preceding claim, wherein the control means is operable to use received data to replace program code used in the 50
14. A coin validator as claimed in any preceding claim.
15. A banknote validator as claimed in any one of claims 1 to 13. 55
16. A method of re-programming a money validator as claimed in any one of claims 1 to 13.
17. A method of re-configuring a money validator as claimed in any one of claims 2 to 11, comprising issuing an instruction to transfer control to the boot program and then supplying data using the communications routine.
18. A method of re-programming a money validator having a plurality of communications ports each capable of receiving re-programming data, the method comprising selecting one of the communication ports, causing the validator to execute a communications routine, and transferring data to the validator such that the data is received by the communications routine and can thus be used to replace existing memory contents of the validator defining the configuration of the validator.
19. A method as claimed in claim 18, wherein the step of causing the validator to execute a communications routine comprises issuing a command using the selected communications port.
20. A method as claimed in any one of claims 17 to 19, including transmitting to the validator data defining acceptance criteria.

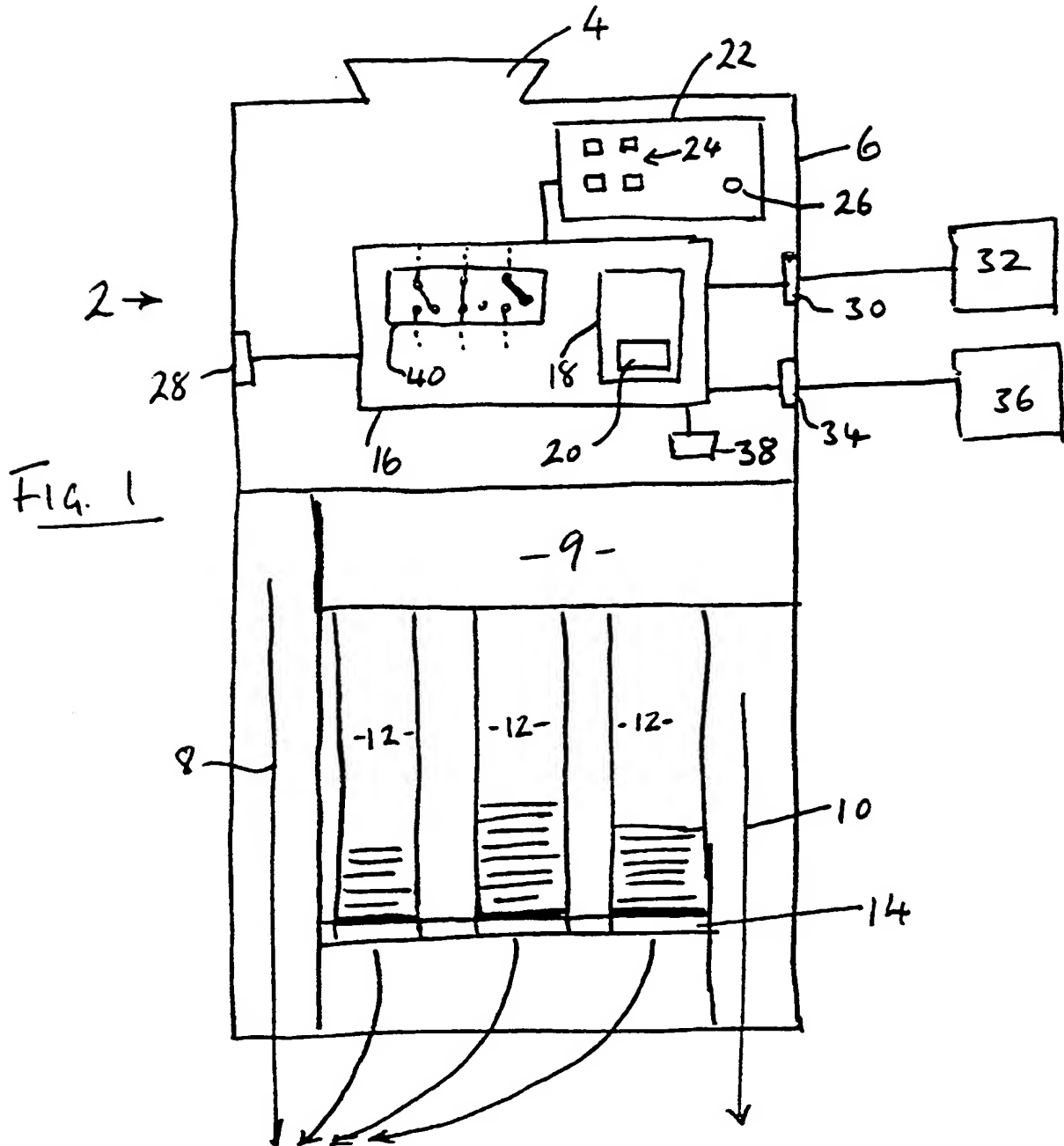
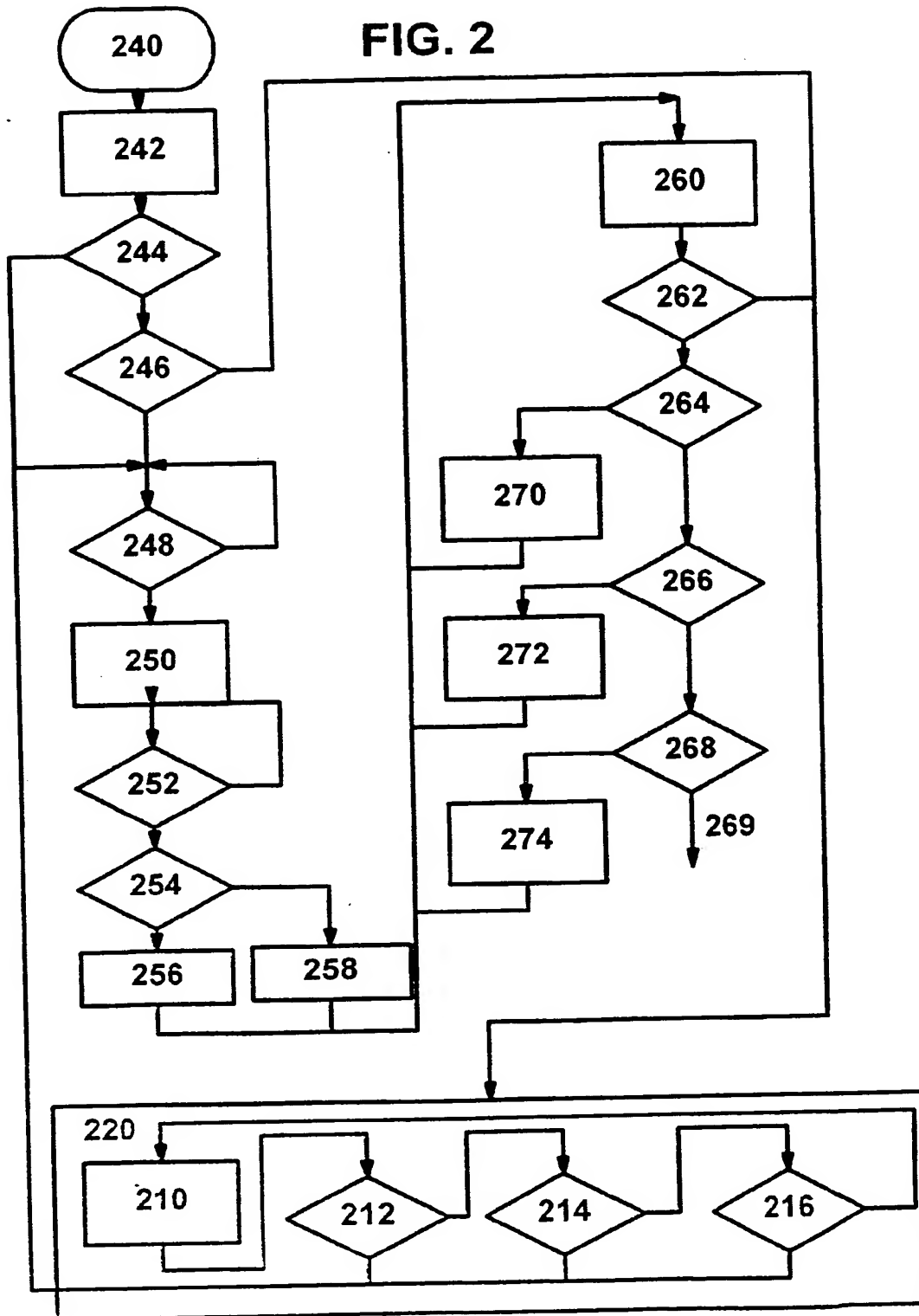


FIG. 2







European Patent  
Office

## EUROPEAN SEARCH REPORT

Application Number  
EP 00 30 2916

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X A	US 5 438 186 A (NAIR PARAMESWARAN B ET AL) 1 August 1995 (1995-08-01) * column 22, line 34 - column 23, line 19 * * column 25, line 11 - line 57 * * column 27, line 38 - line 50 * * column 32, line 36 - line 66 *	1-8, 10-20 9	G07D5/00 G07D7/00
X A	WO 98 19276 A (OHYA TOITSU ;KUROIWA KOJI (JP); NIPPON KINSEN KIKAI KK (JP)) 7 May 1998 (1998-05-07) * abstract *  * page 3, paragraph 2 - paragraph 5 * * page 5, paragraph 2 - page 6, paragraph 2 * * page 10, paragraph 1 - page 13, paragraph 2 * * figures 3-5 *	1,12-16  2-11, 17-20	
X A	WO 97 27567 A (COIN CONTROLS ;BARSON ANDREW WILLIAM (GB)) 31 July 1997 (1997-07-31) * page 4, line 16 - page 5, line 6 *  * page 14, line 11 - page 15, line 12 * * figure 2 *	1,12-16  2-11, 17-20	TECHNICAL FIELDS SEARCHED (Int.Cl.7)  G07D G07F
X A	US 5 341 408 A (MELCHER RICHARD A ET AL) 23 August 1994 (1994-08-23) * column 7, line 7 - column 8, line 49 *  * column 10, line 15 - line 26 * * figure 13 *	1,11-16  2-10, 17-20	
The present search report has been drawn up for all claims			
Place of search <b>THE HAGUE</b>		Date of completion of the search <b>13 July 2000</b>	Examiner <b>Van Dop, E</b>
<p>CATEGORY OF CITED DOCUMENTS</p> <p>X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document</p> <p>T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons &amp; : member of the same patent family, corresponding document</p>			

EPO FORM 1503 03 82 (Pdc01)

**ANNEX TO THE EUROPEAN SEARCH REPORT  
ON EUROPEAN PATENT APPLICATION NO.**

EP 00 30 2916

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.  
The members are as contained in the European Patent Office EDP file on  
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

13-07-2000

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5438186 A	01-08-1995	AU 5538494 A	24-05-1994
		WO 9410649 A	11-05-1994
		US 5448047 A	05-09-1995
		US 5444616 A	22-08-1995
WO 9819276 A	07-05-1998	JP 10134221 A	22-05-1998
		AU 695903 B	27-08-1998
		AU 4726897 A	22-05-1998
		CA 2233646 A	30-04-1998
		EP 0888594 A	07-01-1999
		NO 982925 A	23-06-1998
		ZA 9709806 A	10-09-1998
WO 9727567 A	31-07-1997	AU 7702696 A	20-08-1997
		CA 2243155 A	31-07-1997
		CN 1207815 A	10-02-1999
		DE 69607393 D	27-04-2000
		EP 0876657 A	11-11-1998
US 5341408 A	23-08-1994	MX 9204362 A	01-01-1993

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82